UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/541,399	03/31/2000	Youfeng Wu	884.217US1	3990

21186 7590 05/07/2003

SCHWEGMAN, LUNDBERG, WOESSNER & KLUTH, P.A.
P.O. BOX 2938
MINNEAPOLIS, MN 55402

EXAMINER
WOOD, WILLIAM H

ART UNIT	PAPER NUMBER
2124	6

DATE MAILED: 05/07/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

PRG

Office Action Summary	Application No.	Applicant(s)	
	09/541,399	WU ET AL.	
	Examiner	Art Unit	
	William H. Wood	2124	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *10 February 2003* .

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-39* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-39* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

Claims 1-39 have been examined.

Claim Rejections - 35 USC § 102

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

2. Claims 13, 15-16, 18-21 and 24-25 are rejected under 35 U.S.C. 102(b) as being anticipated by Muchnick, "Advanced Compiler Design Implementation", (pages 169-177).

In regard to claim 13, Muchnick disclosed the limitations:

i) *computer-implemented method* (page 169, first paragraph, global indicates composing hierarchy of flow graphs as presented on page 177, Figure 7.8)

ii) *providing a control flow graph of a program, the graph having an inner region and an outer region* (page 169, first paragraph, global indicates composing hierarchy of flow graphs as presented on page 177, Figure 7.8)

iii) *selecting a representative entry node for the inner region* (page 172, section 7.1; in particular page 174, last two paragraphs; additionally page 621-622, Figures 19.10 and 19.11)

iv) *replacing the inner region with the representative entry node* (page 172, section 7.1; in particular page 174, last two paragraphs)

v) *for each prolog node of the inner region, adding an edge from the prolog node to the representative entry node* (page 172, section 7.1; in particular page 174, last two paragraphs)

vi) *for each epilog node of the inner region, adding an edge from the representative entry node to the epilog node* (page 172, section 7.1; in particular page 174, last two paragraphs)

In regard to claim 15, Muchnick disclosed the limitations:

i) *creating a region source node of the outer region* (page 172, section 7.1; in particular page 174, last two paragraphs; page 177, Figure 7.8)

ii) *for each entry node of the outer region, adding an edge from the region source node to the entry node* (page 172, section 7.1; in particular page 174, last two paragraphs; page 177, Figure 7.8)

iii) *creating a region sink node for the outer region* (page 172, section 7.1; in particular page 174, last two paragraphs; page 177, Figure 7.8)

iv) *for each exit node of the outer region, adding an edge from the exit node to the region sink node* (page 172, section 7.1; in particular page 174, last two paragraphs; page 177, Figure 7.8)

In regard to claim 16, Muchnick disclosed the limitation *wherein the control flow graph includes a plurality of inner regions, and the actions of the method are applied for each of the plurality of inner regions, such that a different augmented control flow graph is created for each of the plurality of inner regions* (inherent, as profiling only one region would not give complete results).

In regard to claim 18, Muchnick disclosed the limitations:

i) *method of augmenting a control flow graph in support of profiling a hierarchical path within a software function* (page 169, first paragraph, global indicates composing hierarchy of flow graphs as presented on page 177, Figure 7.8)

ii) *selecting a representative path within an inner region of a software function, the representative path being identified by a representative entry node and a representative exit node* (page 177, Figure 7.8)

iii) *for each prolog node of the inner region, adding an edge from the prolog node to the representative entry node* (page 172, section 7.1; in particular page 174, last two paragraphs)

iv) *for each epilog node of the inner region, adding an edge from the representative exit node to the epilog node* (page 172, section 7.1; in particular page 174, last two paragraphs)

In regard to claim 19, Muchnick disclosed the limitations:

i) *removing any edges from prolog nodes of the inner region to entry nodes of the inner region other than the representative entry node* (page 174, last two paragraphs and page 175, first paragraph)

ii) *removing any edges from exit nodes of the inner region other than the representative exit node to epilog nodes of the inner region* (page 174, last two paragraphs and page 175, first paragraph)

In regard to claim 20, Muchnick disclosed the limitations:

i) *wherein the software function has a function entry and a function exit, and the inner region has at least one entry node and at least one exit node* (page 169, first paragraph; page 177, Figure 7.8)

ii) *adding an edge from the function entry to each of the at least one entry node of the inner region* (page 169, first paragraph; page 177, Figure 7.8)

iii) *adding an edge from each of the at least one exit node of the inner region to the function exit* (page 169, first paragraph; page 177, Figure 7.8)

In regard to claim 21, Muchnick disclosed the limitation *wherein the control flow graph includes a plurality of inner regions, and the actions of the method are applied for each of the plurality of inner regions* (inherent, as profiling only one region would not give complete results).

In regard to claims 24 and 25, the limitations are substantially the same as the limitations of claims 18 and 20 and therefore rejected in the same manner here.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1-12 and 26-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ball et al., "Efficient Path Profiling" in view of Adl-Tabatabai (USPN 6,170,083).

In regard to claim 1, Ball disclosed the following limitations:

i) initializing an inner path sum (Ball: Section 3, Path Profiling of DAGs)

ii) summing edge values encountered in the inner region with the inner path sum (Ball: Section 3, Path Profiling of DAGs)

iii) when exiting the inner region, modifying a profile indicator that represents the frequency of execution of a path within the inner region (Ball: Section 3, Path Profiling of DAGs)

Ball did not teach the limitations of having an inner and outer path sum and the hierarchical path profiler. Adl-Tabatabai demonstrated that it was known at the time of invention to divide path profiling into a hierarchical nature and thus have

an inner and outer path sum (Adl-Tabatabai: Figure 5; column 4, lines 1-25; column 4, lines 26-50). Furthermore, Adl-Tabatabai demonstrated *after entering an inner region, saving an outer path sum* (necessary for hierarchical code regions for returning after computing one nested level down) and *restoring the outer path sum* (necessary for hierarchical code regions for returning after computing one nested level down). It would have been obvious to one of ordinary skill in the art at the time of invention to implement Ball's efficient path profiling with hierarchy of inner and outer path sums as found in Adl-Tabatabai's teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to provide an efficient profiling mechanism with the functionality of a popular programming language, Java, in order to produce better, faster programs.

In regard to claim 2, Ball and Adl-Tabatabai disclosed the limitation *wherein initializing an inner path sum comprises initializing the inner path sum to a value corresponding to an edge from a region source node to an entry node of the inner region* (inherent in Ball and Adl-Tabatabai's combination that a hierarchy of profiling would initialize a region (the inner path region) to the start of that region).

In regard to claim 3, Ball disclosed the limitation *wherein modifying a profile indicator comprises indexing into an array of profile indicators using the inner path sum* (Ball: page 2, first bulleted item, and left column second full paragraph).

In regard to claim 4, Ball disclosed the limitation *wherein the array of profile indicators is dedicated to paths of the inner region* (Ball: taken alone indicates this in the fact that it only has one region, the inner most).

In regard to claim 5, Ball and Adl-Tabatabai disclosed the limitation *wherein initializing an inner path sum comprises initializing the inner path sum to a value corresponding to an edge from a function entry to an entry node of the inner region* (inherent in Ball and Adl-Tabatabai's combination that a hierarchy of profiling would initialize a region (the inner path region) to the start of that region).

In regard to claim 6, Ball disclosed the limitation *wherein modifying a profile indicator comprises indexing into an array of profile indicators using the inner path sum* (Ball: page 47, first bulleted item).

In regard to claim 7, Ball disclosed the limitation *wherein the array of profile indicators includes profile indicators corresponding to paths in the inner region and profile indicators corresponding to paths outside the inner region* (Ball and Adl-Tabatabia combo indicate this since a hierarchy of different profile paths now exist and therefore an array for all of them is now necessary).

In regard to claim 8, Ball disclosed the limitation *wherein the profile indicator includes a profile counter* (Ball: page 47, first bulleted item), *and modifying the*

profile indicator comprises generating a counter address as a function of the inner path sum (Ball: page 47, first bulleted item).

In regard to claim 9, Ball disclosed the limitation *wherein the inner region includes a plurality of paths, each having an inner path sum corresponding thereto, the inner path sums corresponding to the plurality of paths in the inner region being unique relative to each other* (Ball: page 47, Section 1.1, Algorithm Overview).

In regard to claim 10, Ball and Adl-Tabatabai disclosed the limitation *wherein the inner region is one of a plurality of inner regions* (Adl-Tabatabai: Figure 5; column 4, lines 1-25), *and the inner path sums of the inner region are unique relative to inner path sums corresponding to other inner regions* (Ball and Adl-Tabatabai combo indicates this, without uniqueness profiling would have little use).

In regard to claim 11, Ball and Adl-Tabatabai did not teach *wherein saving an outer path sum comprises pushing the out path sum onto a stack.* Official Notice is taken that at the time of invention, it was known to implement structures of nesting, such as hierarchical structures, with pushing and popping from a stack a value of one level up or down. It would have been obvious to one of ordinary skill in the art at the time of invention to implement Ball and Adl-Tabatabai's profiling with a stack to push and pop the data of a previous level as was known in the art

at the time of invention. This implementation would have been obvious because one of ordinary skill in the art would be motivated to save a separate and distinct value of a previous level for use when processing at the current level reached completion in order to enhance flexibility.

In regard to claim 12, Ball and Adl-Tabatabai did not explicitly teach the limitation *wherein restoring the outer path sum comprises popping the outer path sum from the stack*. Official Notice is taken that at the time of invention, it was known to implement structures of nesting, such as hierarchical structures, with pushing and popping from a stack a value of one level up or down. It would have been obvious to one of ordinary skill in the art at the time of invention to implement Ball and Adl-Tabatabai's profiling with a stack to push and pop the data of a previous level as was known in the art at the time of invention. This implementation would have been obvious because one of ordinary skill in the art would be motivated to save a separate and distinct value of a previous level for use when processing at the current level reached completion in order to enhance flexibility.

In regard to claims 26, Ball and Adl-Tabatabai disclosed the limitations:

i) *computer-implemented method for instrumenting software in support of hierarchical path profiling*

ii) *at an entry to an inner region of a hierarchical software path , inserting an instruction to save an outer path sum*

iii) *within the inner region, inserting instructions to sum edge values into an inner path sum*

iv) *after an exit from the inner region, inserting an instruction to increment a counter addressed as a function of the inner path sum*

The rejection is the same as for claim 1 here since the limitations are the same.

In regard to claim 27, Ball and Adl-Tabatabai disclosed the limitation *after the exit from the inner region, inserting an instruction to restore the outer path sum* in claim 1. The rejection is the same.

In regard to claim 28, Ball and Adl-Tabatabai disclosed the limitation wherein the inner region is one of a plurality of inner regions in a hierarchical arrangement, and the actions of the method are applied to each of the plurality of inner regions (inherent, as profiling only one region would not give complete results).

In regard to claims 29 and 30, the limitations are the same as for claims 26 and 27 and therefore are rejected in the same manner.

5. Claims 14, 23, 31-36 and 39 are rejected under 35 U.S.C. 103(a) as being unpatentable over Muchnick, "Advanced Compiler Design Implementation" as applied to claims 13 and 20 above, respectively, and in view of Ball et al., "Efficient Path Profiling".

In regard to claim 14, Muchnick did not explicitly teach the limitation *assigning edge values to all edges in the control flow graph such that the sum of the edge values along each unique path is unique within the control flow graph*. Ball demonstrated that it was known at the time of invention to assign edge values to produce a unique sum for each path (page 50, section 3.2). It would have been obvious to one of ordinary skill in the art at the time of invention to implement Muchnick's general knowledge of graphing systems with unique edge sum's for profiling as found in Ball's teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to utilize well-known compiler theory with as efficient of a profiler as possible.

In regard to claim 23, Muchnick did not explicitly teach the limitation *assigning edge values to all edges in the control flow graph such that the sum of the edge values along each unique path is unique within the control flow graph*. Ball demonstrated that it was known at the time of invention to assign edge values to produce a unique sum for each path (page 50, section 3.2). It would have been obvious to one of ordinary skill in the art at the time of invention to implement Muchnick's general knowledge of graphing systems with unique edge sum's for profiling as found in Ball's teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to utilize well-known compiler theory with as efficient of a profiler as possible.

In regard to claim 31, Muchnick disclosed the limitations:

- *A computer-implemented method comprising:*
  - *defining an outer region of code for a computer program* (page 610, Figures 19.1 and 19.2; a region of code including procedures, this is an outer region);
  - *defining an inner region of the code embedded within the outer region* (page 610, Figures 19.1 and 19.2; procedures would make up the inner regions of code; pages 621-622, Figures 19.10 and 19.11; page 169-175; discussion of analysis of inner regions of code);

Muchnick did not explicitly state the limitations:

- *assigning inner edge values within the inner region such that the inner edge values have different sums for different paths through the nodes of the inner region*
- *assigning outer edge values within the outer region such that the outer edge values have different sums for different paths through the nodes of the outer region, where the inner region is treated as a single node*
- *while executing the outer region; accumulating an outer path sum of the outer edge values;*
- *in connection with entering the inner region, saving the outer path sum, accumulating an inner path sum of the inner edge values;*
- *in connection with exiting the inner region, modifying a profile indicator that represents the frequency of execution of a path within*

the inner region, restoring the outer path sum.

Ball demonstrated that it was known at the time of invention to apply edge values within a region such that the edge values have different sums for different paths through the nodes of the region (pages 4-7, section 3; page 5, first paragraph under section 3.2). It would have been obvious to one of ordinary skill in the art at the time of invention to include with Muchnick's optimizations technique's an additional technique to further optimize sections/procedures/inner of code as found in Ball's teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to use the newest and most advanced techniques for solving the common and persistent problem of code optimization (which Muchnick provides numerous examples of throughout his text).

Considering, Ball demonstrated edge values and path summing as stated above and Muchnick disclosed inner and outer regions of code being optimized (also stated above), it would have been further obvious to one of ordinary skill in the art at the time of invention to apply edge sums to the outer regions (call graph in Muchnick, page 610) such that the edge values have different paths through the nodes of the outer region (concepts of path summing found in Ball and discussed above), wherein the inner region is treated as a single node (call graph in Muchnick, page 610). This implementation would have been obvious because one of ordinary skill in the art would be motivated to optimize larger sections / entire programs of code along with single procedures (this is why Muchnick

includes techniques for interprocedural analysis). Muchnick provides for nesting inner regions within outer regions and then optimizing, Ball simply demonstrates the concept of path summing. It would further be obvious to apply the Ball techniques to Muchnick, just by looking at the graphs shown in both references. The call graph of Muchnick is consistent with Ball's graphs.

Finally, since the combination of Muchnick and Ball has been shown to be obvious, the remaining limitations of the claim can clearly be seen as apparent as common methods to implement the above combination. *While executing the outer region, accumulating an outer path sum of the outer edge values* (accumulating a sum is defined in Ball, and Muchnick shows applying it to an outer region). *In connection with entering the inner region, saving the outer path sum, accumulating an inner path sum of the inner edge values* (Ball shows accumulating a sum, and Muchnick demonstrates the difference between inner and outer regions). Saving the outer sum, when entering the inner region for processing is not explicitly stated. However, considering the combination of references, it would have been obvious to one of ordinary skill in the art at the time of invention to save the outer sum when entering the inner region. This is because often when switching between areas of work, the previous area (outer region) must be saved, especially if it is going to be returned to for further work. *In connection with exiting the inner region, modifying a profile indicator that represents the frequency of execution of a path within the inner region* (as discussed in Ball: page 2, first bulleted item, and left column second full

paragraph), *restoring the outer path sum* (if the sum is saved, presumably returning to that region for processing would still be obvious to restore the sum).

In regard to claim 32, Muchnick and Ball further disclosed the limitations *wherein the code is executable code* (Muchnick: page 607, "program" construction).

In regard to claim 33, Muchnick and Ball further disclosed the limitations *wherein the inner path sum is uniquely determined for different execution paths in the inner region* (Ball: page 5, section 3.2, first paragraph).

In regard to claim 34, Muchnick and Ball further disclosed the limitations *further comprising addressing the profile indicator with the accumulated inner path sum* (Ball: page 2, first bulleted item, and left column second full paragraph).

In regard to claim 35, Muchnick and Ball further disclosed the limitations *further comprising initializing the inner path sum before accumulating it* (Ball: page 6, section 3.4; second paragraph indicates initializing path register in some manner).

In regard to claim 36, Muchnick and Ball further disclosed the limitations *further comprising repeating the operations of claim 31 wherein the outer region is an inner region embedded in a further outer region* (Muchnick: page 622, Figure 19.11).

In regard to claim 39, Muchnick and Ball further disclosed the limitation *A machine-readable medium including instructions to carry out the method of claim 31* (clearly, Muchnick and Ball's concepts, techniques and algorithms exist in media, which a machine will execute, such as a compiler on a hard drive).

6. Claims 17 and 22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Muchnick, "Advanced Compiler Design Implementation" as applied to claims 15 and 20 above, respectively, and in view of Adl-Tabatabai (USPN 6,170,083).

In regard to claims 17, Muchnick did not explicitly state the limitation *wherein the control flow graph includes a hierarchy of inner regions, and the actions of the method are applied recursively to the hierarchy of inner regions, such that a different augmented control flow graph is created for each inner region in the hierarchy of inner regions*. Adl-Tabatabai demonstrated that it was known at the time of invention to construct a hierarchy of recursive code regions (column 4, lines 1-25). It would have been obvious to one of ordinary skill in the art at the time of invention to implement Muchnick's profiling with a hierarchy of recursive code regions as found in Adl-Tabatabai's teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to implement code profiling with the ability to efficiently profile a popular programming language, Java.

In regard to claim 22, Muchnick did not explicitly state the limitation *wherein the control flow graph includes a hierarchy of inner regions, and the actions of the method are applied recursively to the hierarchy of inner regions.* Adl-Tabatabai demonstrated that it was known at the time of invention to construct a hierarchy of recursive code regions (column 4, lines 1-25). It would have been obvious to one of ordinary skill in the art at the time of invention to implement Muchnick's profiling with a hierarchy of recursive code regions as found in Adl-Tabatabai's teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to implement code profiling with the ability to efficiently profile a popular programming language, Java.

7. Claims 37 and 38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Muchnick, "Advanced Compiler Design Implementation" in view of Ball et al., "Efficient Path Profiling" as applied to claim 36 and in view of "Dictionary of Computing" herein referred to as Computing.

In regard to claim 37, Muchnick and Ball did not explicitly state the limitations *wherein saving an outer path sum comprises pushing the outer path sum onto a stack.* Computing demonstrated that it was known at the time of invention to utilize stacks to save information (pages 470-471; in particular "stack processing"). It would have been obvious to one of ordinary skill in the art at the time of invention to implement Muchnick and Ball's optimization technique's with

saving a value to be used later to the stack as found in Computing's teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to use common systems which are easily understood and implemented, furthermore saving the outer path sum is similar to a recursive function, which is also suggested in "stack" in Computing.

In regard to claim 38, Muchnick and Ball did not explicitly state the limitations *wherein restoring the outer path sum comprises popping the outer path sum from a stack*. Computing demonstrated that it was known at the time of invention to utilize stacks to save and restore information (pages 470-471; in particular "stack processing"). It would have been obvious to one of ordinary skill in the art at the time of invention to implement Muchnick and Ball's optimization technique's with saving and then restoring a value to be used later to the stack as found in Computing's teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to use common systems which are easily understood and implemented, furthermore saving the outer path sum is similar to a recursive function, which is also suggested in "stack" in Computing.

Examiner's Response

8. Applicant's arguments with respect to claim 13, filed on 10 February 2003 in the recent response, have been fully considered, but are not persuasive. Applicant argues the Muchnick reference does not fulfill the limitations of replacing an inner region with an entry node and thus creating a new path through the graph. Examiner respectively disagrees, for instance section 7.1

starting on page 172 describes the process of analyzing control-flow graphs. Basic blocks form the nodes, they replace more complex inner regions and edges are placed in a graph to connect basic blocks to entry and exit points. This is exactly what the broadest reasonable interpretation of Applicant's claim 13 accomplishes. It is common knowledge to replace an inner region with a simpler structure in order to perform analysis of complex graphs.

9. Applicant's arguments with respect to claims 14, 17-19 and 22 have been considered through the above statement regarding claim 13.

10. Applicant's arguments with respect to claim 1 have been fully considered, but are not persuasive. Applicant argues the Ball reference only provides for a single sum and the Adl-Tabatabai has no sums at all and that the claimed limitations are not found by the two references. Examiner disagrees since the references in combination must be considered. Applicant does not argue that Ball discloses summing edges for profiling. In view of Adl-Tabatabai, which disclosed a hierarchical path profiling system, it is clear that complex paths can be analyzed in a piecemeal manner, i.e. hierarchical paths. As stated in the previous Office Action, it would have been obvious to one of ordinary skill in the art at the time of invention to implement Ball's efficient path profiling with hierarchy of inner and outer paths. Considering Ball implements sums for at least an inner path, it is clear that in a hierarchical system inner and outer sums must be considered in order to appreciate the overall analysis. As such, the broadest reasonable interpretation of the claim 1 limitations read upon the cited prior art references.

11. Applicant's arguments with respect to claim 2 have been fully considered, but are not persuasive. Applicant argues there is no grounds for inheritance. Examiner maintains that upon the obviousness combination between Ball and Adl-Tabatabai, the system provides for initializing an inner path sum. It is inherent, that in order to accurately reflect a path, which transcends through a large subdivided code (hierarchical) that initialization of the inner region must in some way reflect the entry from the outer region. Furthermore, the nodes of the claim are nothing more than well-known optimization techniques known to compiler design using flow graphs and DAGs (as mentioned by Ball) as noted in "Advanced Compiler Design Implemenation".

12. Applicant's arguments with respect to claim 5 have been fully considered, but are not persuasive. Applicant argues there is no grounds for inheritance. Examiner maintains that upon the obviousness combination between Ball and Adl-Tabatabai, the system provides for initializing an inner path sum. It is inherent, that in order to accurately reflect a path through a large subdivided code (hierarchical) that initialization of the inner region must in some way reflect the entry from the outer region. Furthermore, the nodes of the claim are nothing more than well-known optimization techniques known to compiler design using flow graphs and DAGs (as mentioned by Ball) as noted in "Advanced Compiler Design Implemenation".

13. Applicant's arguments with respect to claim 10 have been fully considered, but are not persuasive. Applicant states applicant's system uses the feature only for global-based profiling. Examiner does not see how this refutes

the necessity for uniqueness. Furthermore, Ball (page 4, line 25, right column indicates the need for uniqueness.

14. Applicant's arguments with respect to claims 26-30 have been fully considered, but are not persuasive and are rejected as stated above in the rejections and examiner maintains the positions held in related claims as above stated in the arguments.

Conclusion

15. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Correspondence Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to William H. Wood whose telephone number is (703)305-3305. The

examiner can normally be reached 7:30am - 5:00pm Monday thru Thursday and 7:30am - 4:00pm every other Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (703)305-9662. The fax phone numbers for the organization where this application or proceeding is assigned are (703)746-7239 for regular communications and (703)746-7238 for After Final communications.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703)305-3900.

William H. Wood
April 30, 2003

TUAN Q. DAM
PRIMARY EXAMINER